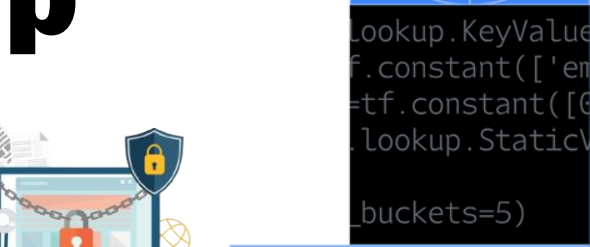
Google Developer Groups

CyberSec Bootcamp Lecture – 0x06



Google Developer Groups

MEET YOUR GUIDES :

Zishan Ansari

(Mentor)

CyberSec - GDG

LinkedIn

GitHub

Google Developer Groups

MEET YOUR GUIDES :

Rohit Choudhary

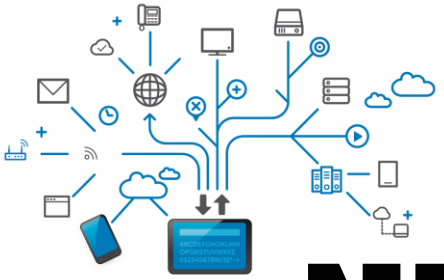
(Instructor)

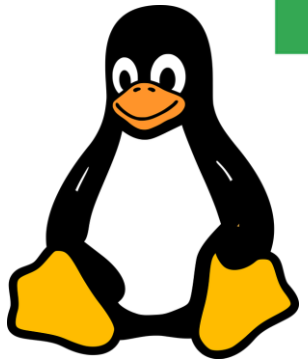
CyberSec - GDG

LinkedIn

GitHub

Google Developer Groups





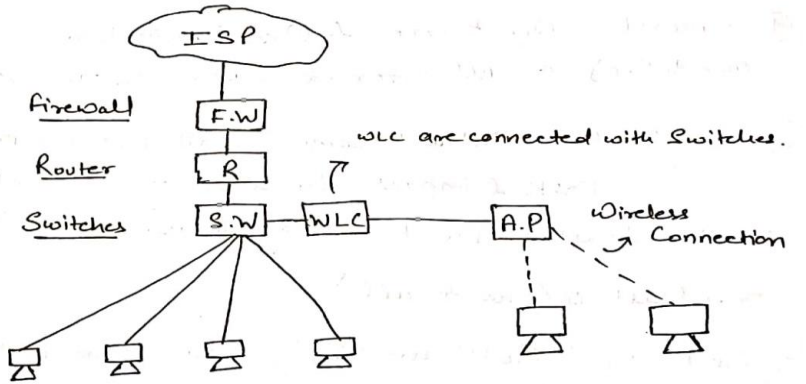
NETWORKING

Google Developer Groups

NETWORKING-

$> Network – Interconnection of Network devices that are used to share data or resources.
Example : Router, Switches, Firewall, Access Points (AP's), Wireless Lan Controller (WLC), End Devices (PC, Server) etc.

$>



Google Developer Groups

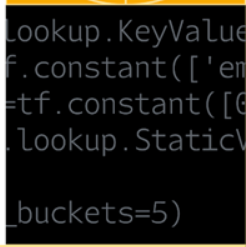
NETWORKING-

$> ISP (Internet Service Provider) : A company that provides internet access to individuals and businesses

$> Switch : Provides centralized location and connect multiple devices in a network

$> Firewall : Protects your network from unauthorized access. Also we can say that it controls (filters) incoming and outgoing traffic in a network on the basis of some rules like company policy

$> Access Point : Provides centralized location to connect devices in a network but without wire using RF signal



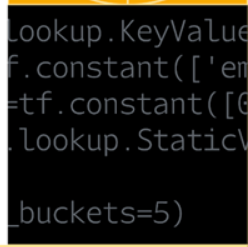
Google Developer Groups

NETWORKING-

$> Wireless LAN Controller (WLC) : Provides centralized management of all access points in a network

$> Network Types :

1. LAN (Local Area Network) – Set of devices connected within a limit geographical area of upto 1KM
2. MAN (Metropolitan Area Network) – Set of devices connected within a city limits MAN Covers a range of upto 50KM (5KM-50KM)
3. WAN (Wide Area Network) – Set of devices connected within wide geographical area. Example : District, State, Country, Continent etc



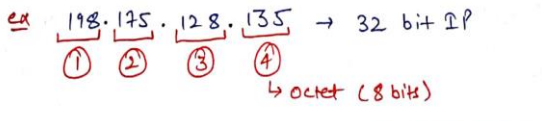
Google Developer Groups

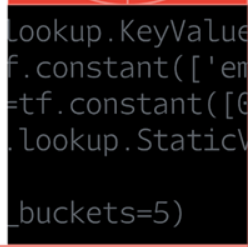
NETWORKING-

$> IP Address (Internet Protocol) : Provides you an unique identity of a Network Device over Internet or in your network. We call it Internal Protocol as it is a protocol of Internet layer

$> Internet ➡ Public IP
In our Network ➡ Private IP

$> IPv4 (Internet Protocol Version 4) : 32 bit identity and is represented in decimal notation. IPV4 is divided into 4 octets

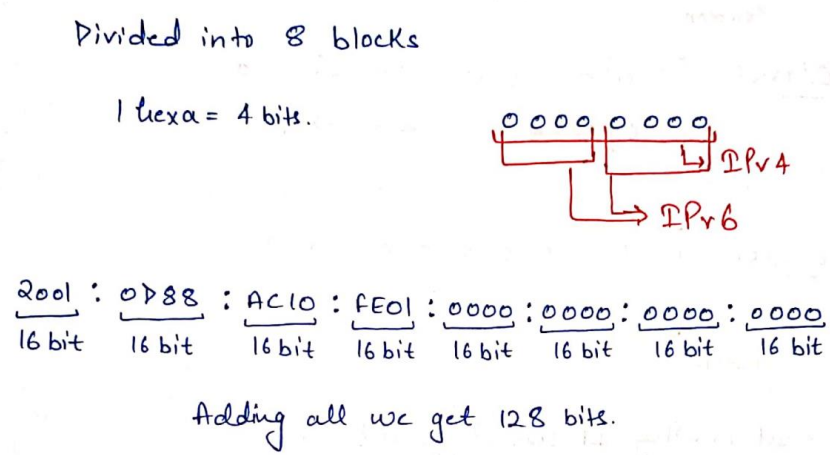




Google Developer Groups

NETWORKING-

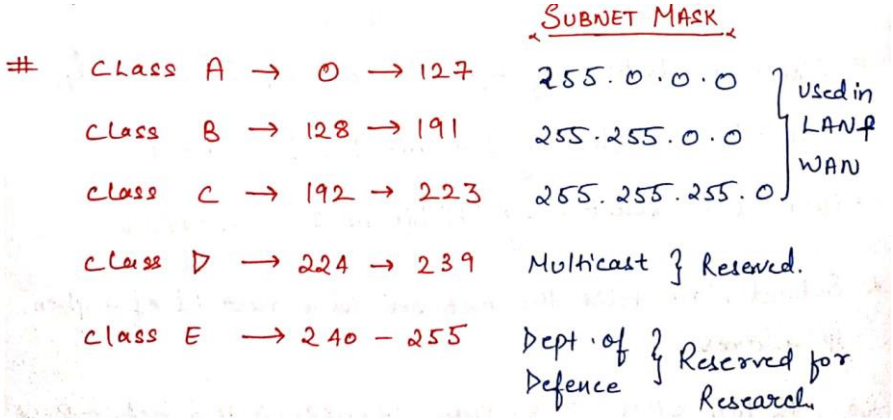
$> IPv6 (Internet Protocol Version 6) : 128 bit identity and represented in hexadecimal

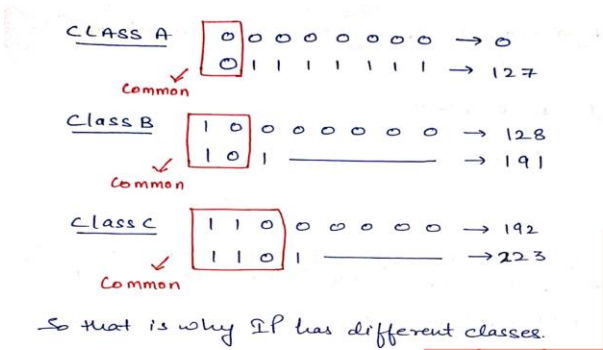


Google Developer Groups

NETWORKING-

$> Internet Protocol Classes : IP Address classes are a system for categorizing IP Addresses based on the size of the network and the number of devices it supports





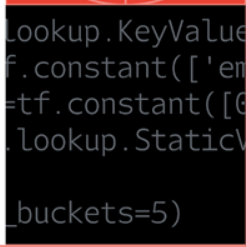
Google Developer Groups

NETWORKING-

$> 0.0.0.0 is a Reserved block/IP which we cannot configure on our PC and it is used in Default Routing or In DHCP during DHCP discover Packet (we will learn about DHCP in Protocol section) and 255.255.255.255 is a Broadcasting IP

$> IP is divided into two parts based on working behavior

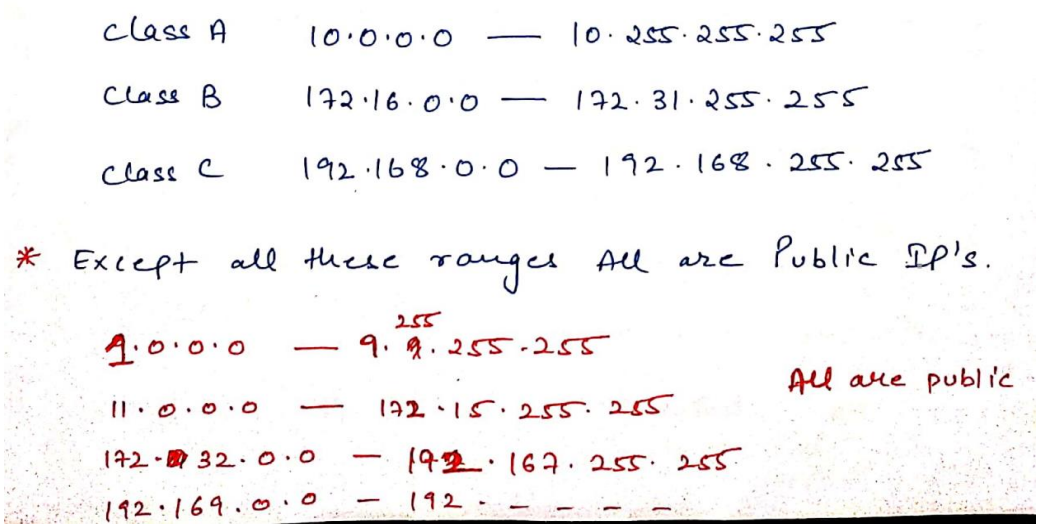
Private IP	Public IP
1. Provided by Network Admin	1. Provided by ISP
2. Locally Unique	2. Globally Unique
3. Used for Internal Communication LAN	3. Used for Internet WAN
4. Free	4. Paid
5. Unregistered	5. Registered



Google Developer Groups

NETWORKING-

$> Range of Private and Public IP :



Google Developer Groups

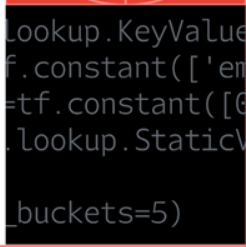
NETWORKING-

$> MAC (Media Access Control) Address : MAC Addresses are unique 48-bit hardware numbers of a computer that are embedded into a network card (NIC) during manufacturing

$> It is a 12-digit hexadecimal number (48-bit binary number), which is mostly represented by Colon-Hexadecimal notation

$> Types of MAC Address:

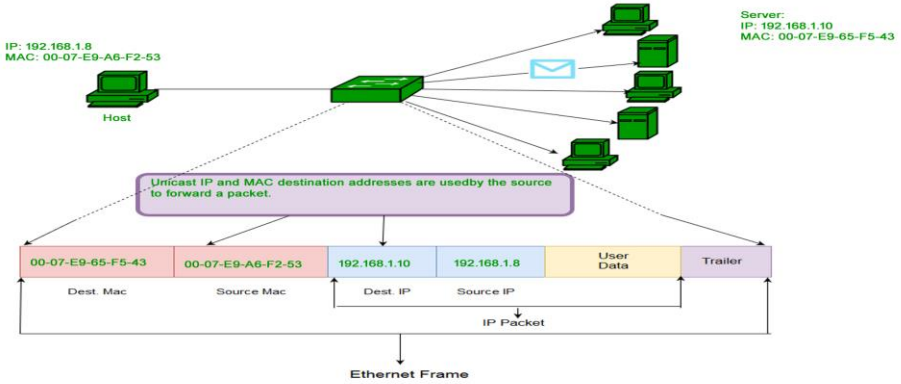
- Unicast
- Multicast
- Broadcast



Google Developer Groups

NETWORKING-

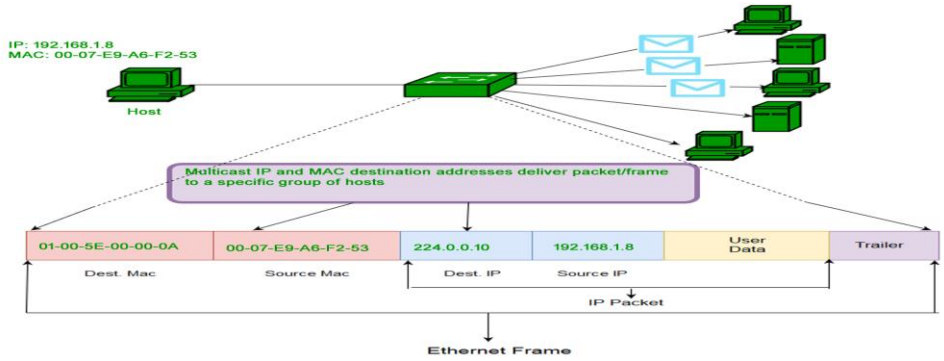
- Unicast : A Unicast-addressed frame is only sent out to the interface leading to a specific NIC. If the LSB (least significant bit) of the first octet of an address is set to zero, the frame is meant to reach only one receiving NIC. The MAC Address of the source machine is always Unicast



Google Developer Groups

NETWORKING-

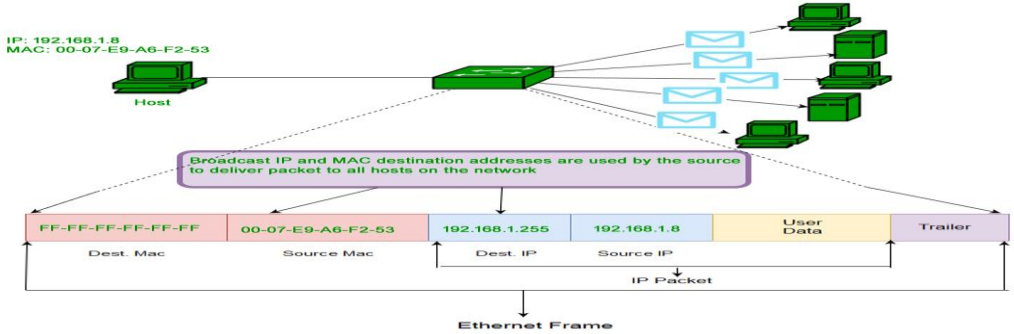
- Multicast : The Multicast Address allows the source to send a frame to a group of devices. In Layer-2 (Ethernet) Multicast Address, the LSB (least significant bit) of the first octet of an address is set to one. IEEE has allocated the address block 01-80-C2-xx-xx-xx (00-00-00 to FF-FF-FF) for group addresses for use by standard protocols



Google Developer Groups

NETWORKING-

- Broadcast : Similar to Network Layer, Broadcast is also possible on the underlying layer (Data Link Layer). Ethernet frames with ones in all bits of the destination address (FF-FF-FF-FF-FF-FF) are referred to as the broadcast addresses. Frames that are destined With MAC Address FF-FF-FF-FF-FF-FF will reach every computer belonging to that LAN



Google Developer Groups

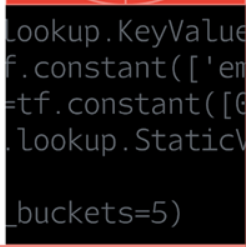
NETWORKING-

$> SUBNET MASK : 32 bit Identity, SUBNET MASK is a mixture of Network ID and Host ID

- Network ID is denoted by ON bit i.e. 1 (fix)
- Host ID is denoted by OFF bit i.e. 1 (variable)

SUBNET MASK tells the Network ID and Host ID of a given IP Address
Example – 192.168.50.1, here 192.168.50 is a Network ID and .1 is a Host ID

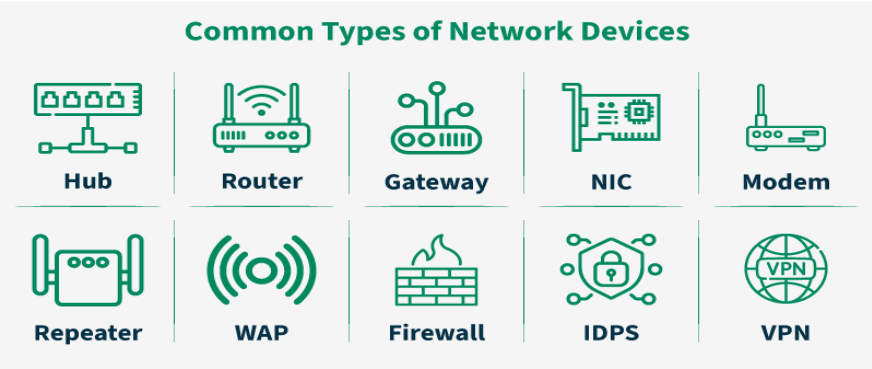
$> SUBNETTING : A method where we divide a large or complex network into small network and that small network is called subnetwork



Google Developer Groups

NETWORKING-

$> Networking Devices : Network devices are physical devices that allow hardware on a computer network to communicate and interact with each other. They ensure efficient communication between connected devices by controlling data transfer, boosting signals, and linking different networks

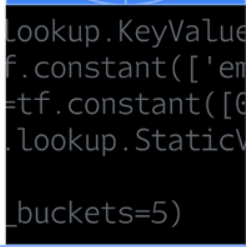


Google Developer Groups

NETWORKING-

- Modem : Modem is also known as modulator/demodulator is a network device that is used to convert digital signal into analog signal of different frequencies and transmits these signals to a modem at the receiving location, also used to convert analog to digital signal

- Repeater : A repeater operates at the physical layer (We will learn more in OSI model), its main function is to amplify (i.e. regenerate) the signal over the same network before the signal becomes too weak or corrupted to extend the length to which the signal can be transmitted over the same network. It copy the signal bit by bit and regenerate it.
It is a 2-port device

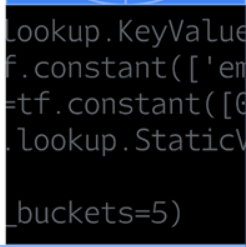


Google Developer Groups

NETWORKING-

➢ Hub : A hub is a multiport repeater. A hub connects multiple wires coming from different branches. Hubs cannot filter data, so packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Types of HUB -

- Active Hub – These are the hubs that have their power supply and can clean, boost, and relay the same signal along with the network
- Passive Hub – These are the hubs that collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them
- Intelligent Hub – It works like an active hub and includes remote management Capabilities. They also provide flexible data rates to network devices.



Google Developer Groups

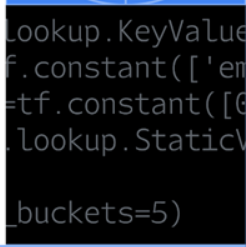
NETWORKING-

- Bridge : A bridge operates at the data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC address of the source and destination. It is also used for interconnecting two LANs working on the same protocol.

- Router : A router is a device like a switch that routes data packets based on their IP

- Gateway : Is a passage to connect two networks that may work upon different networking models

- Network Interface Card (NIC) : A network adapter that is used to connect the computer to the network



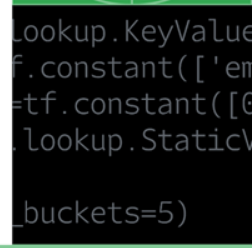
Google Developer Groups

NETWORKING-

$> Topologies : Network Topology is the way to devices are connected in a Network. It defines how these components are connected and how data transfer between the network

$> Types of Network Topology : Below mentioned are the types of Network Topology

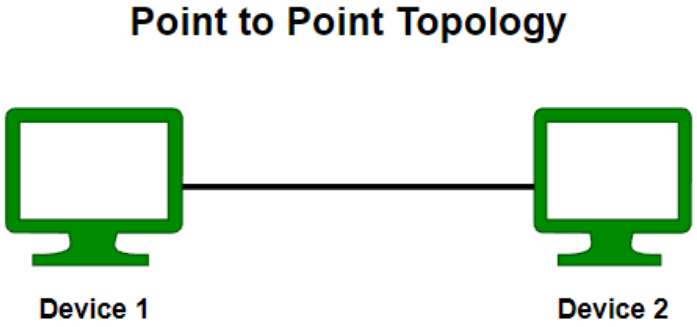
- Point to Point Topology
- Mesh Topology
- Star Topology
- Bus Topology
- Ring Topology
- Tree Topology
- Hybrid Topology



Google Developer Groups

NETWORKING-

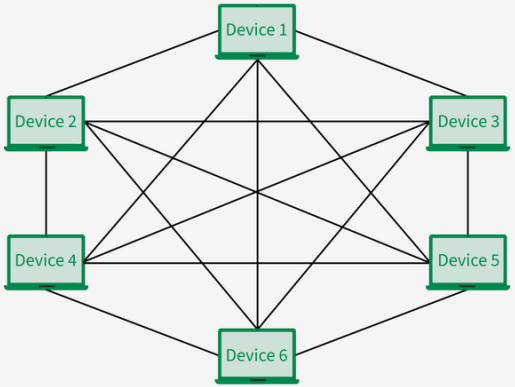
❖ Point to Point Topology : Is a type of topology that works on the functionality of the sender and receiver. It is the simplest communication between two nodes, in which one is the sender and the other one is the receiver. Point-to-Point provides high bandwith



Google Developer Groups

NETWORKING-

❖ Mesh Topology : Every device is connected to another device via particular channel. These channels are known as links. The protocols used are AHCP, DHCP, etc (More about them in Protocol slide)



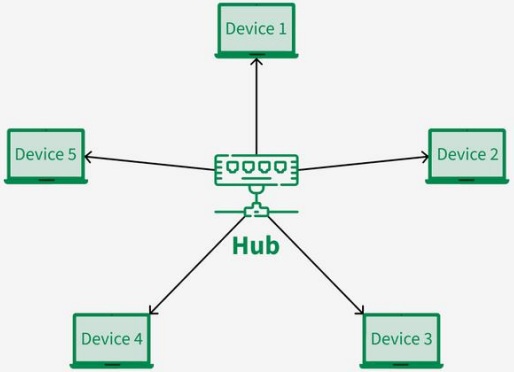
Advantages –
1.Communication is very fast between the nodes
2.Is a robust
3.Fault is diagnosed easily, data is reliable
4.Provides security and Privacy

Disadvantages –
1.Installation and configuration is difficult
2.Expensive

Google Developer Groups

NETWORKING-

❖ Star Topology : All the devices are connected to a single HUB through a cable. This hub is central node and all other nodes are connected to the central node, can be a intelligent or non-intelligent hub

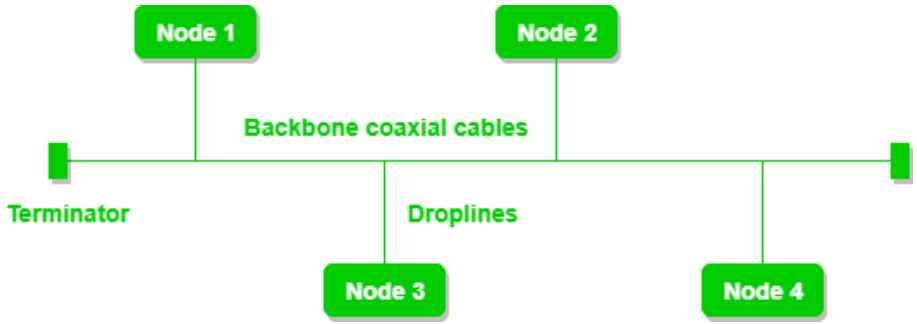


Advantages –
1.Easy to setup
2.It is robust
3.Easu to fault detection and fault isolation
4.Cost-effective, as it used inexpensive coaxial cable
Disadvantages –
1.The whole system depends on HUB
2.Expensive

Google Developer Groups

NETWORKING-

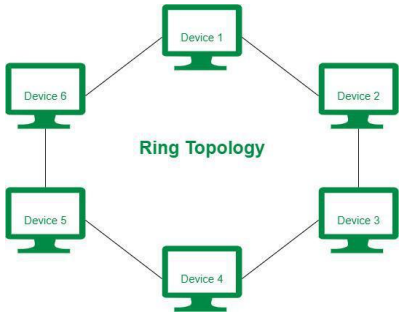
❖ Bus Topology : Every computer and network devices is connected to a single cable. It is bi-directional. It is a multi-Point connection and a non-robust topology because if the backbone fails the topology crashes. Security is very low. Requires a lot of cabling. Adding New devices to the network will slow down networks.



Google Developer Groups

NETWORKING-

❖ Ring Topology : It forms a ring connecting devices with exactly two neighboring devices. A number of repeaters are used in this topology with a large number of nodes, because if someone wants to send some data to the last node in the ring topology with 100 nodes, then the data will have to pass through 99 nodes to reach the 100th node, Hence to prevent data loss repeaters are used in the network. Failure of a single node can cause whole failure



Advantages -

1. The data transmission is high speed
2. The possibility of collision is minimum in this type of topology
3. Cheap to install and expand
4. It is less costly than a star topology

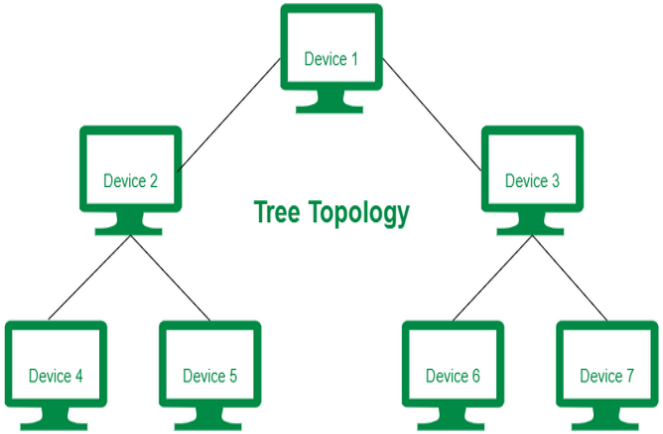
Disadvantages –

1. Troubleshooting is difficult, Less secure

Google Developer Groups

NETWORKING-

❖ Tree Topology : It is a variation of the Star topology. This topology follows a hierarchical flow. Here, various secondary hubs are connected to central hub which contains a repeater. This data flow is from top to bottom. It is multi-point connection and a non-robust topology



Advantages -

1. Error detection and correction are very easy
2. Allows network to get isolated and prioritize from different computers

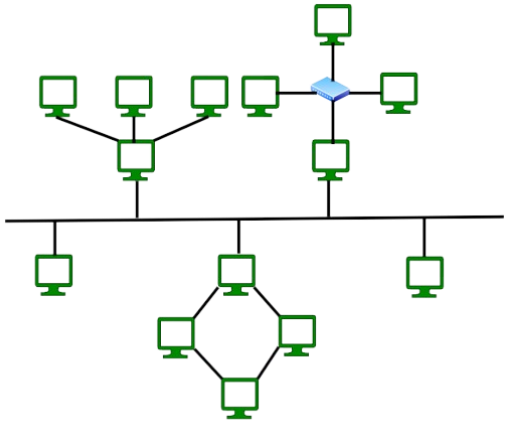
Disadvantages -

1. If the central hub fails, entire sys. fails
2. Cost is high

Google Developer Groups

NETWORKING-

❖ Hybrid Topology : Is the combination of all the various types of topologies, we have studied above. Is used when nodes are free to take any form. It means these can be individuals such as Ring or Star topology, or can be a combination of various types.



Advantages -

1. This topology is very flexible
2. This size of the network can be easily expanded

Disadvantages –

1. It is challenging to design the architecture
2. Hubs used in this topology are very expensive
3. Cost is very high

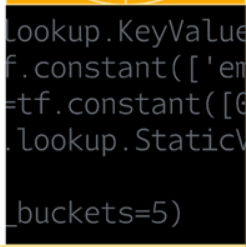
Google Developer Groups

NETWORKING-

$> Protocols : Network Protocols are a set of rules that are responsible for the communication of data between various devices in the network

$> Hypertext Transfer Protocol (HTTP) : It is a layer 7 protocol that is designed for transferring a hypertext between two or more systems. HTTP works on a client-server model, most of the data sharing over the web is done through using HTTP

$> Transmission Control Protocol (TCP) : it establishes a connection between applications before sending any data. It is used for communicating over a network



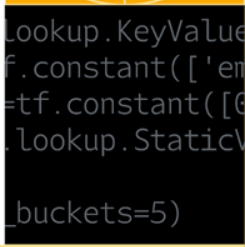
Google Developer Groups

NETWORKING-

$> User Datagram Protocol (UDP) : It is a connectionless protocol that lay-out a basic but unreliable message service. It adds no flow control, reliability, or error-recovery functions. UPD is functional in cases where reliability is not required. It is used when we want faster transmission, for multicasting and broadcasting connections, etc

$> Dynamic Host Configuration Protocol (DHCP) : It's a protocol for network management and it's used for the method of automating the process of configuring devices on IP networks

$> Ad-Hoc Configuration Protocol (AHCP) : AHCP is an autoconfiguration protocol for IPv6 and dual-stack IPv6/IPv4 networks



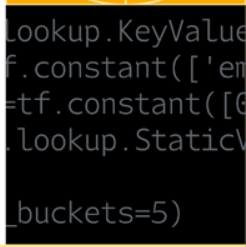
Google Developer Groups

NETWORKING-

$> Internet Control Message Protocol (ICMP) : It is a layer 3 protocol that is used by network devices to forward operational information and error messages. ICMP is used for reporting congestions, network errors, diagnostic purposes, and timeouts

$> File Transfer Protocol (FTP) : FTP is a Client/server protocol that is used for moving files to or from a host computer, it allows users to download files, programs, web pages, and other things that are available on other services

$> Simple Mail Transfer Protocol (SMTP) : Protocol for sending and relaying emails across servers.



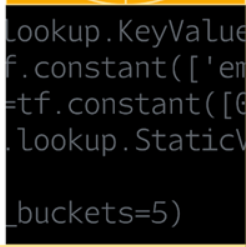
Google Developer Groups

NETWORKING-

$> Voice Over Internet Protocol (VOIP) : Protocol for voice and multimedia communication over the internet

$> Post Office Protocol (POP3) : It is a protocol that a local mail client uses to get email messages from a remote email server over a TCP/IP connection

$> TELNET (Teletype Network) : It is a protocol that allows the user to connect to a remote computer program and to use it i.e., it is designed for remote connectivity. Telnet creates a connection between a host machine and a remote endpoint to enable a remote session



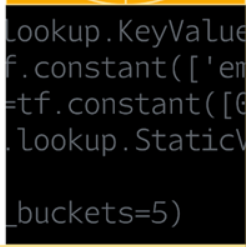
Google Developer Groups

NETWORKING-

$> Secure Shell (SSH) : A protocol that allows users to securely connect to remote servers

$> Secure Socket Layer (SSL) : It is a network security protocol mainly used for protecting sensitive data and securing internet connections. SSL allows both server-to-server and client-to-server communication

$> Hypertext Transfer Protocol (HTTPS) : It is the secured version of HTTP. this protocol ensures secure communication between two computers where one sends the request through the browser and the other fetches the data from the web server

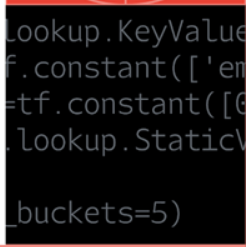


Google Developer Groups

NETWORKING-

$> Port : A port is a virtual point where network connections start and end. Ports are software-based and managed by a computer's operating system

$> Port Number : Ports are standardized across all network-connected devices, with each port assigned a number. Most ports are reserved for certain protocols — for example, all Hypertext Transfer Protocol (HTTP) messages go to port 80. While IP addresses enable messages to go to and from specific devices, port numbers allow targeting of specific services or applications within those devices

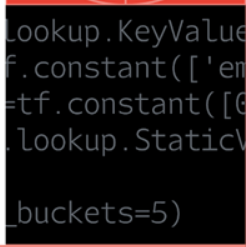


Google Developer Groups

NETWORKING-

$> Different Port Numbers : There are 65,535 possible port numbers, although not all are in common use. Some of the most commonly used ports, along with their associated networking protocol, are –

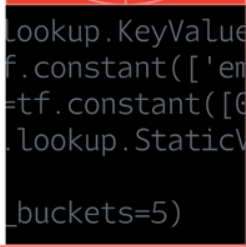
- Port 20 and 21 : File Transfer Protocol (FTP)
- Port 22 : Secure Shell (SSH)
- Port 25 : Simple Mail Transfer Protocol (SMTP)
- Port 53 : Domain Name System (DNS)
- Port 80 : Hypertext Transfer Protocol (HTTP)
- Port 123 : Network Time Protocol (NTP)
- Port 179 : Border Gateway Protocol (BGP)



Google Developer Groups

NETWORKING-

- Port 443 : HTTP Secure (HTTPS)
- Port 500 : Internet Security Association and Key Management Protocol (ISAKMP)
- Port 587: Modern, secure SMTP that uses encryption
- Port 3389 : Remote Desktop Protocol (RDP)



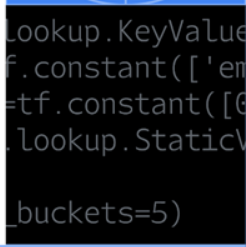
Google Developer Groups

NETWORKING-

$> Open System Interconnection (OSI) Model : a standardized model which we use to demonstrate the theory behind computer networking. In practice, it's actually the more compact TCP/IP model that real-world networking is based off; however the OSI model, in many ways, is easier to get an initial understanding from

The OSI Model Consists of Seven Layers –

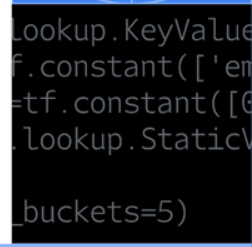
7. Application Layer
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer



Google Developer Groups

NETWORKING-

$> There are many mnemonics floating around to help you learn the layers of the OSI model search around until you find one that you like.
I personally favor: Anxious Pale Shakespeare Treated Nervous Drunks Patiently



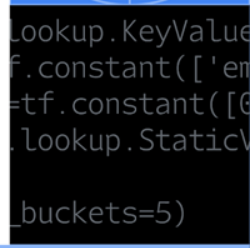
❖ Layer 7 – Application :
The application layer of the OSI model essentially provides networking options to programs running on a computer. It works almost exclusively with applications, providing an interface for them to use in order to transmit data. When data is given to the application layer, it is passed down into the presentation layer

Google Developer Groups

NETWORKING-

❖ Layer 6 – Presentation :

The presentation layer receives data from the application layer. This data tends to be in a format that the application understands, but it's not necessarily in a standardized format that could be understood by the application layer in the receiving computer. The presentation layer translates the data into a standardized format, as well as handling any encryption, compression or other transformations to the data. With this complete, the data is passed down to the session layer

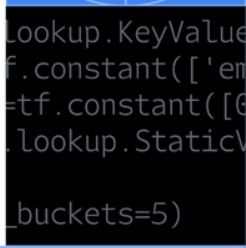


Google Developer Groups

NETWORKING-

❖ Layer 5 – Session :

When the session layer receives the correctly formatted data from the presentation layer, it looks to see if it can set up a connection with the other computer across the network. If it can't then it sends back an error and the process goes no further. If a session can be established then it's the job of the session layer to maintain it, as well as co-operate with the session layer of the remote computer in order to synchronize communications. When the session layer has successfully logged a connection between the host and remote computer the data is passed down to Layer 4 – The transport Layer

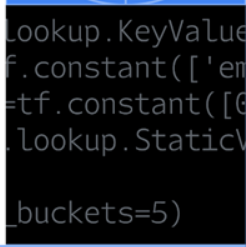


Google Developer Groups

NETWORKING-

❖ Layer 4 – Transport :

The transport layer is a very interesting layer that serves numerous important functions. Its first purpose is to choose the protocol over which the data is to be transmitted. The two most common protocols in the transport layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). With a protocol selected, the transport layer then divides the transmission up into bite-sized pieces (over TCP these are called segments, over UDP they're called datagrams), which makes it easier to transmit the message successfully

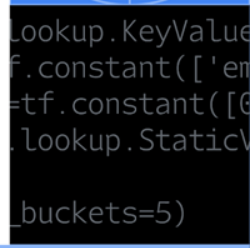


Google Developer Groups

NETWORKING-

❖ Layer 3 – Network :

The network layer is responsible for locating the destination of your request. For example, the Internet is a huge network; when you want to request information from a webpage, it's the network layer that takes the IP address for the page and figures out the best route to take. At this stage we're working with what is referred to as Logical addressing (i.e. IP addresses) which are still software controlled. Logical addresses are used to provide order to networks, categorizing them and allowing us to properly sort them. Currently the most common form of logical addressing is the IPV4 format

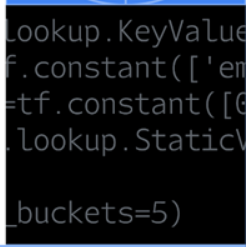


Google Developer Groups

NETWORKING-

❖ Layer 2 – Data Link :

The data link layer focuses on the physical addressing of the transmission. It receives a packet from the network layer (that includes the IP address for the remote computer) and adds in the physical (MAC) address of the receiving endpoint. Inside every network enabled computer is a Network Interface Card (NIC) which comes with a unique MAC address to identify it. Additionally, it's also the job of the data link layer to present the data in a format suitable for transmission. The data link layer also serves an important function when it receives data, as it checks the received information to make sure that it hasn't been corrupted during transmission, which could well happen when the data is transmitted by layer 1: the physical layer

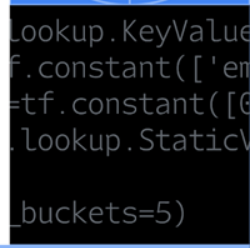


Google Developer Groups

NETWORKING-

❖ Layer 1 – Physical :

The physical layer is right down to the hardware of the computer. This is where the electrical pulses that make up data transfer over a network are sent and received. It's the job of the physical layer to convert the binary data of the transmission into signals and transmit them across the network, as well as receiving incoming signals and converting them back into binary data



Google Developer Groups

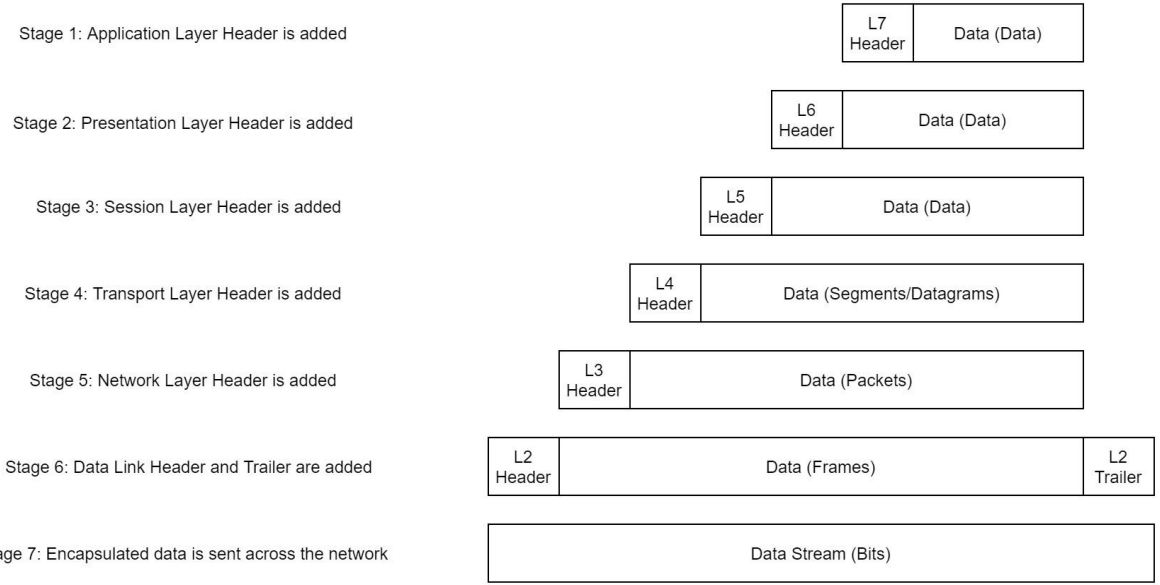
NETWORKING-

$> Encapsulation : As the data is passed down each layer of the model, more information containing details specific to the layer in question is added on to the start of the transmission. As an example, the header added by the Network Layer would include things like the source and destination IP addresses, and the header added by the Transport Layer would include (amongst other things) information specific to the protocol being used. The data link layer also adds a piece on at the end of the transmission, which is used to verify that the data has not been corrupted on transmission; this also has the added bonus of increased security, as the data can't be intercepted and tampered with without breaking the trailer. This whole process is referred to as encapsulation; the process by which data can be sent from one computer to another.

Google Developer Groups

NETWORKING-

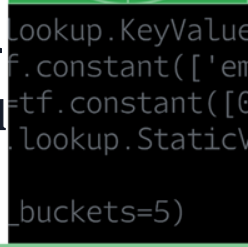
$>



Google Developer Groups

NETWORKING-

$> Notice that the encapsulated data is given a different name at different steps of the process. When the message is received by the second computer, it reverses the process starting at the physical layer and working up until it reaches the application layer, stripping off the added information as it goes. This is referred to as de-encapsulation. The processes of encapsulation and de-encapsulation are very important -- not least because of their practical use, but also because they give us a standardized method for sending data

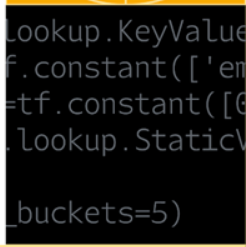


Google Developer Groups

NETWORKING-

$> TCP/IP Model : The TCP/IP model is, in many ways, very similar to the OSI model. It's a few years older, and serves as the basis for real-world networking. The TCP/IP model consists of four layers: Application, Transport, Internet and Network Interface. Between them, these cover the same range of functions as the seven layers of the OSI Model.

- Application Layer
- Transport Layer
- Internet Layer
- Network Interface Layer



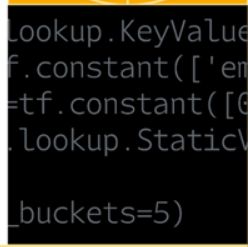
Google Developer Groups

NETWORKING-

$> You would be justified in asking why we bother with the OSI model if it's not actually used for anything in the real-world. The answer to that question is quite simply that the OSI model (due to being less condensed and more rigid than the TCP/IP model) tends to be easier for learning the initial theory of networking.
The two models match up something like this :

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	

$> The processes of encapsulation and de-encapsulation work in exactly the same way with the TCP/IP model as they do with the OSI model. At each layer of the TCP/IP model a header is added during encapsulation, and removed during de-encapsulation.



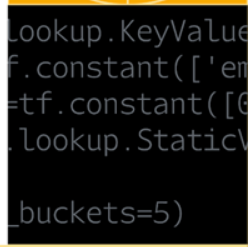
Google Developer Groups

NETWORKING-

$> You would be justified in asking why we bother with the OSI model if it's not actually used for anything in the real-world. The answer to that question is quite simply that the OSI model (due to being less condensed and more rigid than the TCP/IP model) tends to be easier for learning the initial theory of networking.
The two models match up something like this :

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	

$> The processes of encapsulation and de-encapsulation work in exactly the same way with the TCP/IP model as they do with the OSI model. At each layer of the TCP/IP model a header is added during encapsulation, and removed during de-encapsulation.

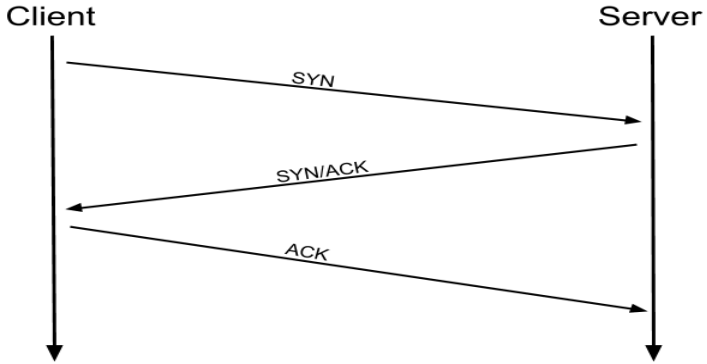


Google Developer Groups

NETWORKING-

$> Three-way Handshake : TCP is a connection-based protocol. In other words, before you send any data via TCP, you must first form a stable connection between the two computers. The process of forming this connection is called the three-way handshake

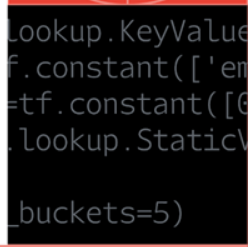
$>



Google Developer Groups

NETWORKING-

$> When you attempt to make a connection, your computer first sends a special request to the remote server indicating that it wants to initialize a connection. This request contains something called a SYN (short for synchronize) bit, which essentially makes first contact in starting the connection process. The server will then respond with a packet containing the SYN bit, as well as another "acknowledgement" bit, called ACK. Finally, your computer will send a packet that contains the ACK bit by itself, confirming that the connection has been setup successfully. With the three-way handshake successfully completed, data can be reliably transmitted between the two computers. Any data that is lost or corrupted on transmission is re-sent, thus leading to a connection which appears to be lossless

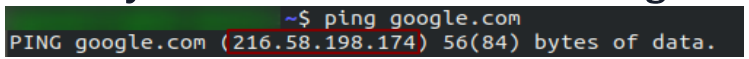


Google Developer Groups

NETWORKING-

$> Networking Tools :

❖ ping Command – The ping command is used when we want to test whether a connection to a remote resource is possible. Usually this will be a website on the internet, but it could also be for a computer on your home network if you want to check if it's configured Correctly. Ping works using the ICMP Protocol



❖ traceroute Command – Traceroute can be used to map the path your request takes as it heads to the target machine

Google Developer Groups

NETWORKING-

$>

```
                    ~$ traceroute google.com
traceroute to google.com (216.58.205.46), 30 hops max, 60 byte packets
 1  _gateway (172.16.255.254)  14.883 ms  15.401 ms  15.551 ms
 2  193.60.160.253 (193.60.160.253)  1.464 ms  1.872 ms  2.026 ms
 3  193.60.168.92 (193.60.168.92)  3.084 ms  4.093 ms  4.814 ms
 4  ge-0-3-2.dund-ban1.ja.net (146.97.128.85)  4.768 ms  4.253 ms  4.715 ms
 5  ae1.dund-ban3.ja.net (146.97.64.97)  10.320 ms  5.114 ms  10.589 ms
 6  ae24.leedaq-sbr2.ja.net (146.97.37.181)  11.160 ms  10.855 ms  10.766 ms
 7  ae29.lowdss-sbr1.ja.net (146.97.33.50)  11.992 ms  11.048 ms  10.746 ms
 8  ae31.londtw-sbr2.ja.net (146.97.33.30)  13.558 ms  13.245 ms  13.561 ms
 9  ae28.londtt-sbr1.ja.net (146.97.33.61)  13.541 ms  13.229 ms  11.410 ms
10  72.14.205.74 (72.14.205.74)  15.143 ms  14.607 ms  13.865 ms
11  74.125.242.97 (74.125.242.97)  13.263 ms 74.125.242.65 (74.125.242.65)  12.553 ms  12.904 ms
12  172.253.71.191 (172.253.71.191)  13.943 ms  12.833 ms 172.253.71.189 (172.253.71.189)  12.631 ms
13  lhr48s23-in-f14.1e100.net (216.58.205.46)  13.227 ms  12.258 ms  12.482 ms
```

$> You can see that it took 13 hops to get from my router (_gateway) to the Google server at 216.58.205.46

Google Developer Groups

NETWORKING-

❖ **whois Command – Whois** is a command-line utility used in **Linux** systems to retrieve information about domain names, IP addresses, and network devices registered with the **Internet Corporation for Assigned Names and Numbers** (ICANN)

$> Notice that we've got the domain name, the company that registered the domain, the last renewal, and when it's next due, and a bunch of information about nameservers

Google Developer Groups

NETWORKING-

❖ **dig Command** – Dig allows us to manually query recursive DNS servers of our choice for information about domains. Command is : dig <domain> @<dns-server-ip>

```
~$ dig google.com @1.1.1.1

; <<>> DiG 9.11.3-1ubuntu1.11-Ubuntu <<>> google.com @1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49715
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1452
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             157     IN      A       216.58.213.14

;; Query time: 22 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sun Mar 08 01:11:47 GMT 2020
;; MSG SIZE  rcvd: 55
```

This information is telling us that we sent it one query and successfully received answer which contains IP address for the domain name. Another interesting piece of information that dig gives us is TTL (time to live) of the queried DNS record. The TTL of the record tells your computer when to stop considering the record as being valid
(it is in seconds)

157

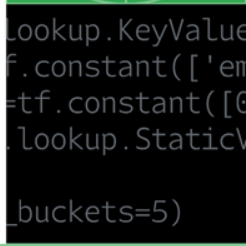
Google Developer Groups

NETWORKING-

$> nmap (Network Mapper) : is an open-source network scanning and security auditing tool that helps users: Discover hosts and services on a network, Identify operating systems and nodes, Find open ports and systems, Understand network devices and services, and Uncover security issues and system loopholes. Nmap can be accessed by typing nmap into the terminal, followed by some “-switches”, for help menu “nmap –h” or “man nmap”

$> Basic Switches of nmap –

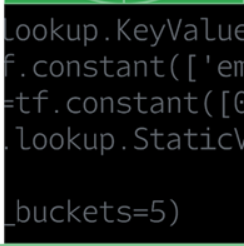
- -sS : for SYN scan, -sT : for TCP scan
- -sU : for UDP scan
- -O : To detect which operating system target is running on
- -sV : To detect version of services running on the target



Google Developer Groups

NETWORKING-

- -v : For verbosity (gives more information on target)
- -vv : For level two (more) verbosity
- -oA : To save the nmap results in three major formats
- -oN : To save the nmap results in a normal format
- -oG : To save results in greppable format
- -A : To enable aggressive mode
- -T5 : nmap offers five level of “timing” template. These are essential to increase the speed of your scan runs at, Note – Higher speeds are noisier and can incur errors
- -p “port number” : To scan a particular port
- -p “range” : To scan a range of ports (e.g. : nmap –p 1000-5000)
- -p- : To scan all ports



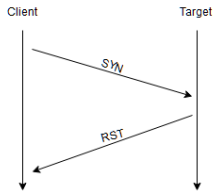
Google Developer Groups

NETWORKING-

- --script : To activate a script from the nmap scripting library
- --script=“category” : To activate all of the scripts in a particular category

(e.g. : nmap --script=vuln)

$> TCP Connect Scans on NMAP : In other words, if Nmap sends a TCP request with the SYN flag set to a closed port, the target server will respond with a TCP packet with the RST (Reset) flag set. By this response, Nmap can establish that the port is closed.

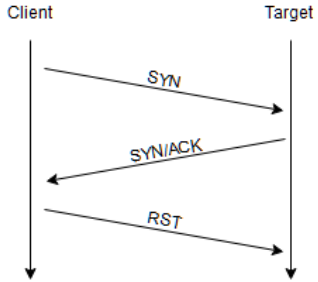


If, however, the request is sent to an open port, the target will respond with a TCP packet with the SYN/ACK flags set. Nmap then marks this port as being open (and completes the handshake by sending back a TCP packet with ACK set)

Google Developer Groups

NETWORKING-

$> SYN Connect Scans on NMAP : SYN scans are sometimes referred to as "Half-open" scans, or "Stealth" scans. Where TCP scans perform a full three-way handshake with the target, SYN scans sends back a RST TCP packet after receiving a SYN/ACK from the server (this prevents the server from repeatedly trying to make the request). In other words, the sequence for scanning an open port looks like this:



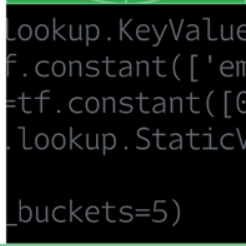
Note : NMAP can use a SYN scan only with Sudo permission

Google Developer Groups

NETWORKING-

$> UDP Connect Scans on NMAP : Unlike TCP, UDP connections are stateless. This means that, rather than initiating a connection with a back-and-forth "handshake", UDP connections rely on sending packets to a target port and essentially hoping that they make it. This makes UDP superb for connections which rely on speed over quality (e.g. video sharing), but the lack of acknowledgement makes UDP significantly more difficult (and much slower) to scan.

When a packet is sent to an UDP port, there should be no response. When this happens, Nmap refers to the port as being open|filtered. In other words, it suspects that the port is open, but it could be firewalled. If it gets a UDP response (which is very unusual), then the port is marked as open. More commonly there is no Response, in which case request is sent a second time as a double-check



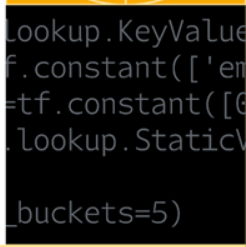
Google Developer Groups

NETWORKING-

$> netcat (nc) : It is a utility tool that uses TCP and UDP connections to read and write in a network. It can be used for both attacking and security. The help command is “nc –h”
To connect to a server the syntax is – nc “target IP” “target Port” “option”

$> Connecting to Telnet : You can connect to a telnet server with the following syntax
telnet “ip” “port”

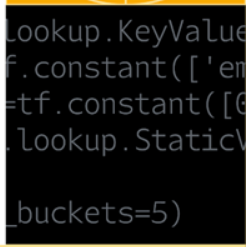
$> Connecting to FTP : You can connect to a FTP server with the following syntax
FTP “ip” “port”
USERNAME
PASSWORD (which we usually crack)



Google Developer Groups

NETWORKING-

$> Connecting to SMTP : same as FTP, SMTP “IP” “PORT” then USERNAME and PASSWORD
Password is obtained by cracking it



Google Developer Groups

THANK YOU